AIAA 2000-0418

Multidisciplinary High-Fidelity Analysis and Optimization of Aerospace Vehicles, Part 1: Formulation

J. L. Walsh, J. C. Townsend, A. O. Salas, J. A. Samareh,
V. Mukhopadhyay, and J.-F. Barthelemy
NASA Langley Research Center
Hampton, VA 23681

38th Aerospace Sciences Meeting & Exhibit
10-13 January 2000 / Reno, NV

MULTIDISCIPLINARY HIGH-FIDELITY ANALYSIS AND OPTIMIZATION OF AEROSPACE VEHICLES, PART 1: FORMULATION

J. L. Walsh,[*] J. C. Townsend,[†] A. O. Salas,[‡] J. A. Samareh,[§] V. Mukhopadhyay,[¶] and J.-F. Barthelemy[**]
NASA Langley Research Center, Hampton, VA 23681

Abstract

An objective of the High Performance Computing and Communication Program at the NASA Langley Research Center is to demonstrate multidisciplinary shape and sizing optimization of a complete aerospace vehicle configuration by using high-fidelity, finite-element structural analysis and computational fluid dynamics aerodynamic analysis in a distributed, heterogeneous computing environment that includes high performance parallel computing. A software system has been designed and implemented to integrate a set of existing discipline analysis codes, some of them computationally intensive, into a distributed computational environment for the design of a high-speed civil transport configuration. The paper describes the engineering aspects of formulating the optimization by integrating these analysis codes and associated interface codes into the system. The discipline codes are integrated by using the Java programming language and a Common Object Request Broker Architecture (CORBA) compliant software product. A companion paper presents currently available results.

Nomenclature

I direction	local fiber (0°) direction
II direction	local transverse to fiber (90°) direction.
σ_x	inplane stress in I direction
σ_y	inplane stress in II direction
τ_{xy}	inplane shear stress
X_C	allowable compressive stress in the I direction
X_T	allowable tensile stress in the I direction
Y_C	allowable compressive stress in the II direction
Y_T	allowable tensile stress in the II direction
S	allowable shear stress in the principal material system
N_{xx}	inplane stress resultant in I direction
N_{yy}	inplane stress resultant in II direction
N_{xy}	inplane shear stress resultant
N_1	largest compressive stress resultant [$-1 * \mathrm{Min}\,(N_{xx}, N_{yy})$]
N_{mn}	biaxial buckling load
N_{Shear}	shear buckling load

Introduction

An objective of the High Performance Computing and Communications Program (HPCCP) at the NASA Langley Research Center (LaRC) has been to promote the use of advanced computing techniques to rapidly solve the problem of multidisciplinary optimization of aerospace vehicles. In 1992, the HPCCP Computational Aerosciences (CAS) team at the LaRC began a multidisciplinary analysis and optimization software development project. Initially, the focus of the CAS project was on the software integration system, or framework, that was used to integrate fast analyses on a simplified design application. The sample application has been the High-Speed Civil Transport (HSCT, Fig. 1). Over the years, the CAS

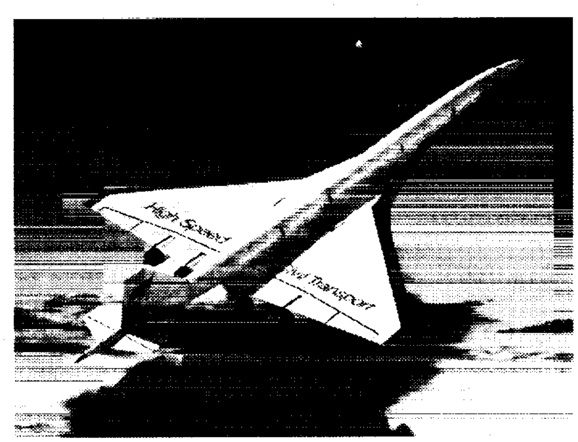


Fig. 1 High-Speed Civil Transport.

[*]Senior Research Engineer, Senior Member AIAA
[†]Senior Research Engineer, Associate Fellow AIAA
[‡]Research Scientist
[§]Research Scientist, Senior Member AIAA
[¶]Senior Research Engineer, Associate Fellow AIAA
[**]Assistant Head Multidisciplinary Optimization Branch, Senior Member AIAA

project has been focused on progressively more complex engineering applications, with the application in the present study known as HSCT4.0. Two previous applications, known as HSCT2.1[1] and HSCT3.5,[2] are briefly summarized next. The HSCT has also been the focus of other research studies (see Refs. 3–9).

The HSCT2.1 application considered a notional wing-only concept and was a multidisciplinary application that integrated very rapid analyses representing aerodynamics, structures, performance, and propulsion. A panel code (WINGDES)[10] with a surface grid having approximately 1000 grid points was used for the aerodynamic analysis. An equivalent laminated plate analysis code (ELAPS)[11] with a structural model having approximately 100 degrees of freedom (DOFs) was used for the structural analysis. The Breguet range equation was used for performance analysis, an engine deck was used for the propulsion analysis, and the only load condition used was that for cruise. The optimization problem consisted of five design variables—two structural design variables (inboard and outboard skin thickness) and three aerodynamic design variables (sweep, root chord, and span at the break)—and required approximately 10 minutes per optimization cycle (analysis, sensitivity, and optimization).

The HSCT3.5 application considered a notional aircraft concept and was a multidisciplinary application that integrated medium-fidelity analyses representing aerodynamics and structures and included rapid performance and propulsion analyses. A marching Euler code (ISAAC)[12] was used with a volume grid having approximately 15,000 grid points for the aerodynamic analysis. A finite-element analysis code (COMET)[13] was used with a finite-element model (FEM) having approximately 15,000 DOFs for the structural analysis. Again, the Breguet range equation was used for performance analysis, an engine deck was used for the propulsion analysis, and the only load condition used was that for cruise. The optimization problem consisted of seven design variables—four structural design variables (inboard and outboard skin thickness distributions) and three aerodynamic design variables (sweep, root chord, and span at the break)—and took approximately 3 hours per optimization cycle (analysis, sensitivity, and optimization).

In 1997, the sample application[14] shifted to more realistic models and higher fidelity analysis codes. This application, known as HSCT4.0, is the focus of this paper. A companion paper[15] discusses the results obtained to date with the implementation of the HSCT4.0 formulation. The HSCT4.0 application objective is to demonstrate simultaneous multidisciplinary shape and sizing optimization of a complete aerospace vehicle configuration by using high-fidelity finite-element structural analysis and computational fluid dynamics (CFD) aerodynamic analysis in a distributed, heterogeneous computing environment that includes high performance parallel computing. To this end, an integrated system of discipline analysis codes and interface codes has been formulated as a distributed computational environment for the design of an HSCT configuration. The analysis part of the design loop has been implemented into a software integration system that is known as CORBA-Java Optimization (CJOpt)[16,17] and is based on a Common Object Request Broker Architecture (CORBA)[18] compliant software product and the Java programming language.

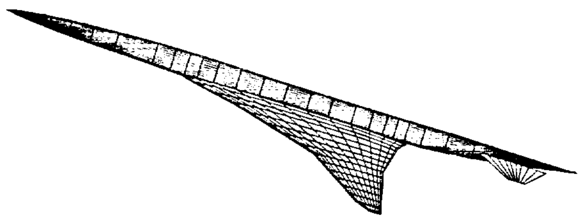
The present paper describes the engineering aspects of formulating the system of discipline analysis codes (some of them computationally intensive) and associated interface codes for integration into CJOpt. First, the HSCT4.0 application, including model definition and optimization problem definition, will be discussed. Next, the HSCT4.0 analysis and formulation will be discussed in terms of processes. Because of the complexity of the project, formal software configuration management is used; so a discussion of the software configuration management experiences with the HSCT4.0 application is included next. Finally, the status of the HSCT4.0 application is summarized. The major analysis codes are described in the appendix. Results are presented in a companion paper.[15]

Overview

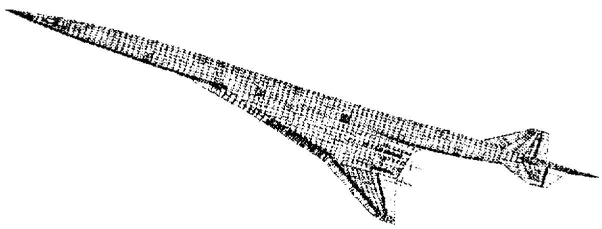
HSCT4.0 Model

The HSCT4.0 application considers a realistic aircraft concept and is a multidisciplinary application that integrates high-fidelity analyses representing aerodynamics, structures, and performance. For the HSCT4.0 application, a realistic model* of an HSCT is used. This model was originally presented in Ref. 19. Other researchers are also investigating the use of multidisciplinary analyses, but with simple generic HSCT models.[3-9] Figure 2 shows both the linear aerodynamics grid and the structural FEM for half of the symmetric baseline HSCT4.0 model. Both a surface grid having approximately 1100 grid points for

* The computational model for this example has been supplied by the Boeing Company and the results are presented without absolute scales in this paper under the conditions of a NASA Langley Property Loan Agreement, Loan Control Number I922931.



a) Linear aerodynamic grid



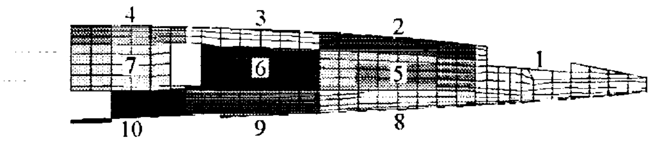
b) Finite-element model

Fig. 2 Baseline HSCT4.0 model.

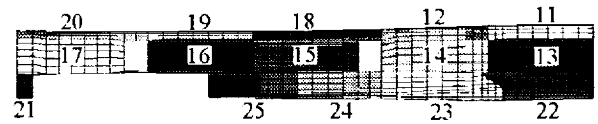
a linear code (USSAERO)[20] and a volume grid having approximately 600,000 grid points for a nonlinear code (CFL3D)[21] are used in combination for the aerodynamic analyses. A FEM with approximately 40,000 DOFs is used with the structural analysis code (GENESIS,®† a product of VMA Engineering).[22] Eight laterally symmetric load conditions are used—one representing a cruise load condition, six arising from those for the maneuver conditions at +2.5g and −1g, and one representing a taxi condition. The performance model is embedded in the Flight Optimization System (FLOPS)[23] code.

Optimization Problem

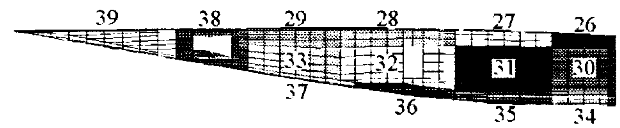
The objective function of the HSCT4.0 optimization problem is to minimize the gross takeoff aircraft weight subject to geometry, structural, performance, and weight constraints. The geometry constraints include constraints on fuel volume, ply mixture ratio, airfoil interior thickness, takeoff ground scrape angle, and landing scrape angle. The structural constraints include buckling and stress constraints. The performance constraints include constraints on range, takeoff field length, landing field length, approach speed, a time-to-climb-to-cruise requirement, and noise. The weight constraints are on operational empty weight and various weight components. The total number of constraints is on the order of 32,000. More detail on the constraints will be given in the next section of the paper including one method of reducing the number of constraints.



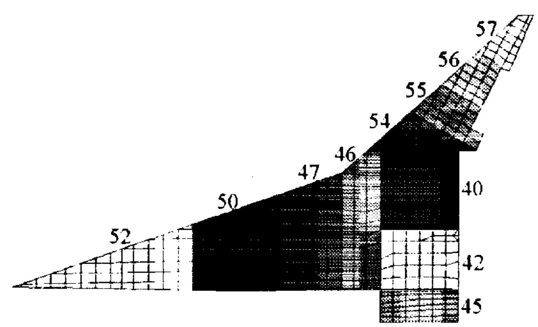
Forward fuselage



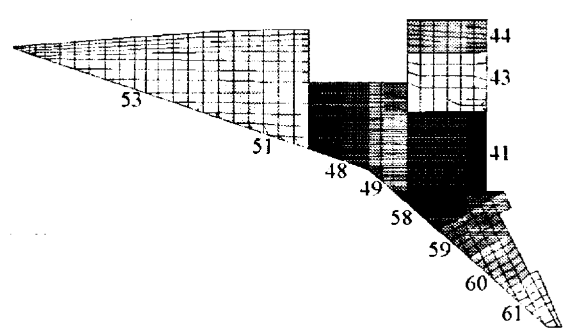
Middle fuselage



Aft fuselage



Upper wing



Lower wing

Fig. 3 Structural design zones.

† The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

The HSCT4.0 application has 271 design variables for optimization—244 structural thickness variables and 27 shape variables. To limit the number of independent structural design variables, the optimization model is divided into 61 design variable zones, as shown in Fig. 3. Each zone consists of several finite elements. Thirty-nine zones are located on the fuselage and 22 zones are located on the wing (11 on the upper surface and 11 on the lower surface). Within each zone, four structural design variables are used. These structural design variables consist of three ply thickness variables (a 0° fiber variable, a 90° fiber variable, and a variable that sizes the 45° and –45° fibers) and a core thickness variable. The composite laminate stacking sequence is shown in Fig. 4.

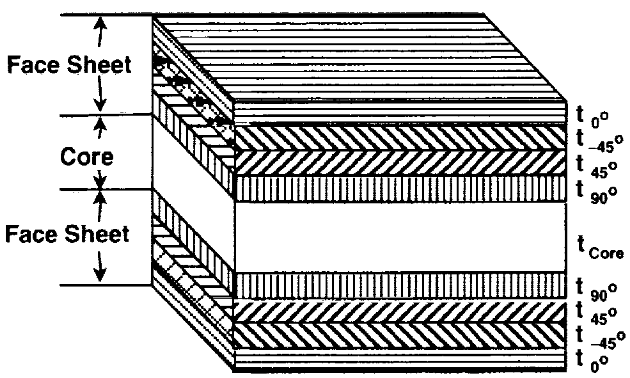
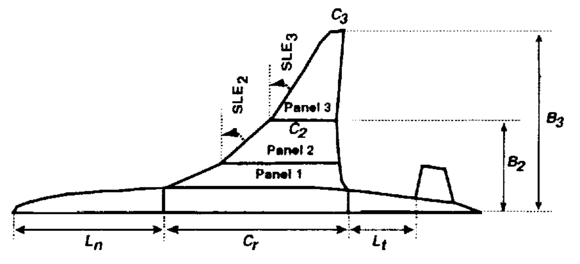
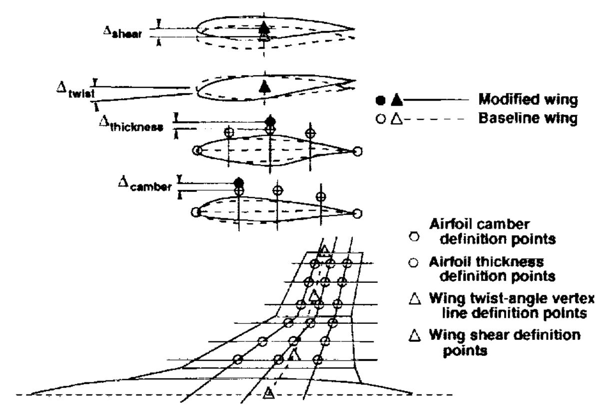


Fig. 4. Composite laminate stacking sequence.



a) Planform design variables.



b) Wing camber, thickness, twist, and shear design variables.

Fig. 5 Shape design variables.

The 27 shape design variables (see Fig. 5) consist of two sets. The first set contains the nine planform variables shown in Fig. 5a—the root chord C_r, the outer break chord C_2, the tip chord C_3, the semispan distance to the outer break B_2, the leading edge sweep of the two outer wing panels SLE_2 and SLE_3, the total projected area of the three wing panels A_t, and the fuselage nose and tail lengths L_n and L_t. Note that the root chord also sets the length of the center fuselage section and that the wing semispan variable B_3 is dependent on other planform variables, including the total area. The second set of shape design variables (see Fig. 5b) consists of control points that define the wing camber, thickness, twist, and shear at a set of airfoil shape definition points. For HSCT4.0, the definition points for camber and thickness are identical and the points for the wing twist line and the wing shear definition are identical. The 18 airfoil shape variables for HSCT4.0 are the vertical (z) perturbations of the camber, thickness, and shear from the wing baseline shape and the wing twist perturbation from the baseline shape in constant y planes. Note that the airfoil camber and thickness perturbations are smooth globally, while the twist and shear perturbations are linear between the line definition points.

HSCT4.0 Analysis and Optimization Formulation

The HSCT4.0 analysis and optimization is formulated in terms of a series of data flow diagrams such as that shown in Fig. 6. These diagrams and an associated set of interface tables show the basic information flow among the analyses. In the diagrams, circles are used to indicate processes (or functions) and arrows show the data that is passed between processes. Not all data passed between processes is explicitly shown, only enough data to indicate the required sequencing among processes. A shaded circle represents a process that is further expanded into a set of processes. For example, the shaded *Analysis* circle in Fig. 6 is further expanded into the 10 processes shown in Fig. 7, each of which can be further expanded. In this paper, all *Analysis* processes in Fig. 7 will be discussed. Detailed diagrams will be presented only for the *Geometry* and *Loads Convergence* processes. By convention, this paper will use italics for process names.

***Optimization* Process**

Figure 6 illustrates the optimization procedure, which consists of a multidisciplinary analysis (*Analysis*), gradient calculations (*Sensitivity Analysis*), and a gradient-based optimizer (*Gradient-Based Optimizer*). The outer loop shown in Fig. 6 represents one design "cycle." A design cycle is defined as analysis (evaluation of the objective function and constraints), sensitivity analysis, and optimization.

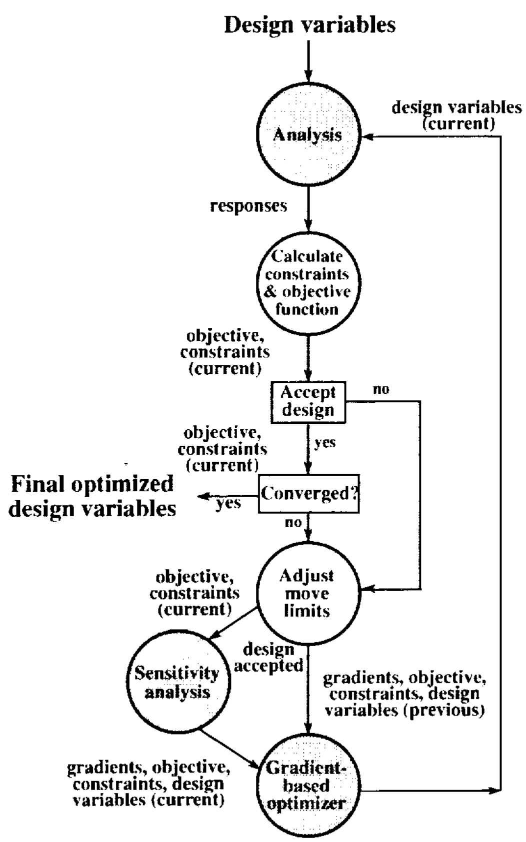


Fig. 6 *Optimization* process.

***Gradient-Based Optimizer* Process**

The *Gradient-Based Optimizer* process, based on a sequential linear programming (SLP) technique, consists of a general-purpose optimization program (CONMIN)[24] and an approximate analysis that is used to reduce the number of full analyses during the optimization procedure. The approximate analysis is used to extrapolate the objective function and constraints with linear Taylor Series expansions. This extrapolation is accomplished by using derivatives of the objective function and constraints (from the *Sensitivity Analysis* process) computed from the analysis at the beginning of each design cycle. Move limits are imposed on the design variables during the *Gradient-Based Optimizer* process to control any errors introduced by the linearity assumption.

***Sensitivity Analysis* Process**

The *Sensitivity Analysis* process provides the derivatives of the constraints and the objective function. Because not every analysis is a direct function of the design variables, it is necessary to obtain the constraint and/or objective function derivatives by chain-ruling component derivatives. The plan is to use analytical derivatives whenever possible, either by hand-differentiating the equations or by using the automatic differentiation tools ADIFOR[25-27] and ADIC,[28] to obtain the component derivatives from any analysis for which source code is available.

The GENESIS® source code is not available. This leads to the major difficulty in obtaining derivatives for the HSCT4.0 application—choosing a method to obtain the total stress and buckling constraint derivatives. The stress and buckling constraints depend on the equilibrium equations for linear static structural analysis:

$$\mathbf{Ku} = \mathbf{f} \tag{1}$$

where $\mathbf{K}$ is the linear stiffness matrix, $\mathbf{u}$ is the vector of nodal displacements, and $\mathbf{f}$ is the applied load vector, which depends on the aeroelastic loads from the *Loads Convergence* process (described later in the paper). The total stress and buckling constraint derivatives depend on component derivatives obtained by differentiating Eq. (1) with respect to a design variable V_i

$$\frac{\partial \mathbf{K}}{\partial \mathbf{V}}\mathbf{u} + \mathbf{K}\frac{\partial \mathbf{u}}{\partial \mathbf{V}} = \frac{\partial \mathbf{f}}{\partial \mathbf{V}} \tag{2}$$

Normally, in structural optimization, it is assumed that constant loads are used, so $\partial \mathbf{f}/\partial V = 0$, and methods exist in the GENESIS® code for obtaining the stress and the buckling constraint derivatives based on that assumption. The plan for the HSCT4.0 project is not to assume constant loads because shape design variables are used. One method is to obtain $\partial \mathbf{f}/\partial V$ by finite differences; this method can be computationally intensive for 271 design variables. An alternate, approximate method to incorporate non-zero $\partial \mathbf{f}/\partial V$ is to exploit the modal approach described in Ref. 8.

***Analysis* Process**

Figure 7 shows a diagram for the HSCT4.0 multidisciplinary analysis process (*Analysis*, Fig. 6). In the HSCT2.1 and HSCT3.5 applications, there were

approximately 10 and 20 processes, respectively. In the HSCT4.0 application, there are approximately 70 instantiations of processes, counting each of the distinct instances in which processes appear in the *Analysis* process; this total does not include repetitive invocations due to iterations.

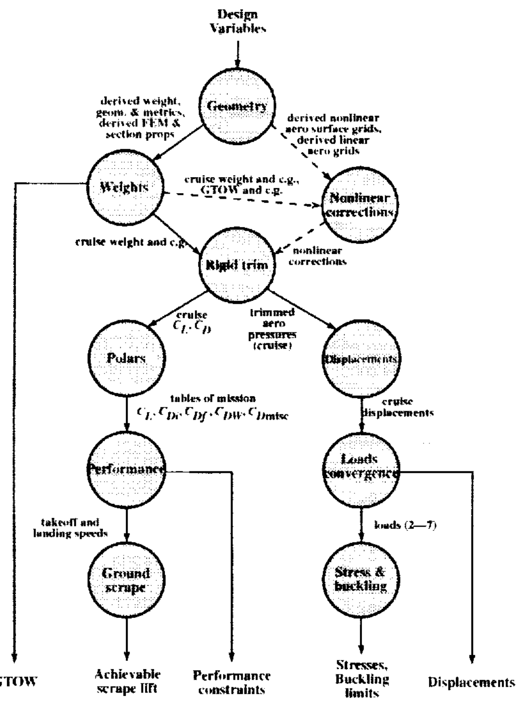


Fig. 7 *Analysis* process.

The *Analysis* process begins at the top when the design variables have been prescribed. First, the *Geometry* process derives updated geometries and grids from baseline geometries and grids for use by later processes. The next step involves using the derived FEM and section properties in a *Weights* process to calculate detailed weights and the center of gravity locations for specified mass cases. The weights data are needed before the remaining processes can be executed. Next, the *Nonlinear Corrections* process can be executed. Note that the flow lines to this process are dashed; the dashed lines indicate that the *Nonlinear Correction* process may not be run in some design cycles due to the high computational time requirements. When this process is not run, the most recent nonlinear corrections continue to be used until an update is available. Next the *Rigid Trim* process is executed to determine the configuration angle of attack and the tail deflection angle that combine to yield a lift equal to the weight, with no net pitching moment for the cruise condition. Once the *Rigid Trim* process has completed, the left branch in Fig. 7, comprising the *Polars*, *Performance*, and *Ground Scrape* processes, can proceed in parallel with the right branch, comprising the *Displacements*, *Loads Convergence*, and *Stress & Buckling* processes; the processes in each branch, however, must proceed sequentially.

Geometry Process

The *Geometry* process provides shape parameterization for the HSCT4.0 application. An important feature of any shape optimization formulation is the means to parameterize the geometry in terms of a set of user-defined design variables that can be systematically varied during the optimization to improve the design. (Reference 29 provides a survey of shape parameterization techniques for multidisciplinary optimization and highlights some emerging ideas.) As shown in Fig. 8, the *Geometry* process consists of 10 processes: *Linear Aero Model Update*, *Nonlinear Aero Surface Model Update*, *Misc Geometry Update*, *FEM Update*, *Performance Geometry*, *Weights Geometry*, *Scrape Geometry*, *Fuel Geometry*, *Section Property Update*, and *Structural Geometry*. Each process is described below.

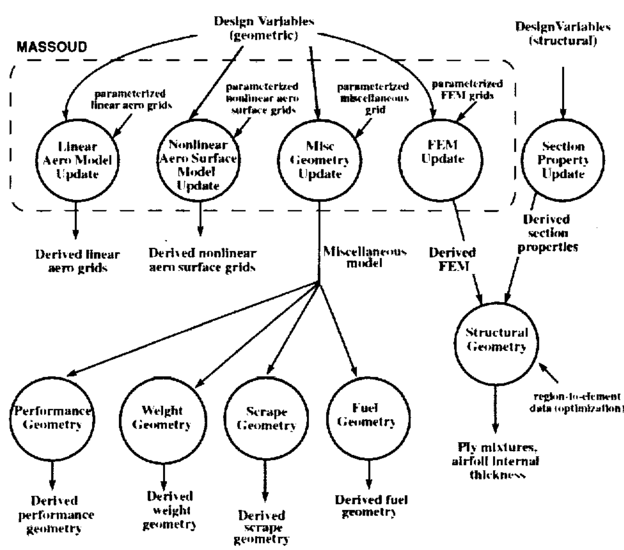


Fig. 8 *Geometry* Process.

The first four geometry processes, shown in Fig. 8, use the MASSOUD (Multidisciplinary Aero/Stru Shape Optimization Using Deformation)[30] code to modify the geometry of the analysis models. The MASSOUD code provides internal FEM grids consistent with aerodynamic surface grids. All analysis geometry models (i.e., aero and structures) are

parameterized based on the locations of design variables that are varied relative to the baseline geometry. The parameterization is done off-line once. For each design cycle, the new derived models are created automatically based on the new set of design variable values. The resulting models are output in the appropriate file formats for subsequent disciplinary analyses. For the linear aerodynamics, nonlinear aerodynamics, and miscellaneous model updates, the MASSOUD code output files represent the new shape in PLOT3D[31] format. The derived linear aerodynamic grids are converted from PLOT3D format to a format more commonly used for linear aerodynamics analyses.

The miscellaneous surface geometry output from the MASSOUD code, shown in Fig. 9, consists of a set of curves that defines a wire-frame description of the model for the various miscellaneous geometry processes. For example, the scrape geometry process reads the location of selected points on the aircraft surface and calculates the pitch angle for which one or more of these points touches the ground. The fuel geometry reads the locations of a set of points at the corners of the fuel tanks and calculates the total and individual fuel volumes of the tanks. The *Weights Geometry* and *Performance Geometry* processes read discretized curves from the miscellaneous geometry files and calculate a wide variety of geometric information needed as input for the *Weights* and *Performance* processes, respectively; examples are the wing span, sweep angles, and aspect ratio, the wing chords and maximum thickness at several span stations, and the fuselage dimensions.

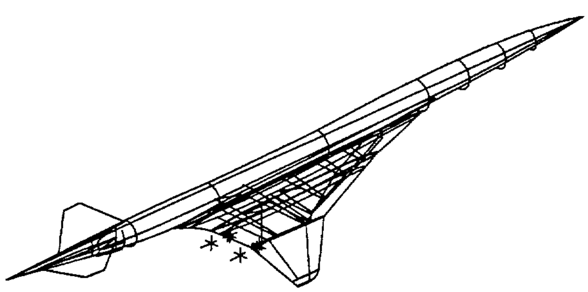


Fig. 9 Miscellaneous geometry.

The remaining processes do not involve the MASSOUD code directly, although the *Structural Geometry* process uses output from the MASSOUD code. The *Section Property Update* process derives the structural section properties from the 244 structural design variables to produce 61 laminated composite shell property data sets in the GENESIS® code format.

The *Structural Geometry* process is used to compute both the ply mixture and airfoil interior thickness constraints. For each ply orientation of the composite face sheets (Fig. 4) of the laminate, constraints were imposed on the ratios of ply thickness to total face sheet laminate thickness. The ply mixture constraints are formulated as follows: the total 0° ply thickness is to make up at least 10 percent of the face sheet laminate thickness, the total 90° ply thickness is to make up at least 10 percent of the face sheet laminate thickness, the total ±45° ply thickness is to make up at least 40 percent but no more than 60 percent of the face sheet laminate thickness. Therefore, 4 ply mixture constraints are used for each of the 61 optimization regions (Fig. 3), for a total of 244 ply mixture constraints.

The airfoil interior thickness (AIT) constraints are computed at each of 30 wing stations (each with a corresponding upper and lower airfoil surface node). The constraint is that the airfoil interior thickness (see Fig. 10) is greater than a specified minimum thickness. The AIT is computed as the distance r between the upper and lower surface nodes minus the average of the upper and lower skin thicknesses (t_{upper} and t_{lower}). The AIT constraint is normalized by the average skin thickness.

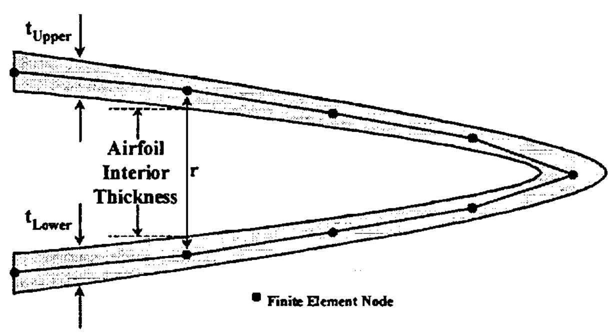


Fig. 10 Airfoil section showing measurements used in Airfoil Interior Thickness constraints

Weights Process

The Weights process computes the as-built nodal weights, component weights, the total configuration weights, and the weight distribution (including the center of gravity location). An attempt is made here to mimic, in a simple way, the functionality of the Boeing as-built weight process, described by Mitchell,[32] without duplicating or including all the process steps and detail of the Boeing as-built weight process. A brief summary of the as-built aircraft weights discussion follows. The as-built weight of a

component includes both the theoretical finite-element model structural weight, plus two kinds of as-built weight increments: 1) weight increments for production splices, local pad-ups, side-of-body joints, adhesives, paints, materials for damage tolerance, sealants, and fasteners essential in building the aircraft and 2) weight increments for remaining items such as windows, landing gear doors, access doors, seat tracks, fuel tank baffles, passenger doors, and system attachment fittings. The total weight of the aircraft can also be thought of as consisting of several weight types: 1) the theoretical finite-element model weight plus the group 1 as-built weight increments above which comprise the as-built structural weight, 2) the non-structural weight which are mostly the group 2 as-built weight increments above, 3) systems weights which include all the various systems normally provided in a working aircraft and which are usually purchased in large quantities by the airframe builders from independent distributors (for example, avionics, auxiliary power, hydraulics, electrical, fuel, passenger accommodation, anti-icing, and air conditioning systems), 4) payload, and 5) fuel. Of the modeled finite-element structure, primary structure (for example, the inboard/outboard wing and forward/mid/aft fuselage structure) is that which is sized directly by configuration or structural design variables, whereas secondary structure (horizontal/vertical tails, engine struts, nose cone, and control surfaces) changes size and weight only as needed to remain consistent (through design variable linking) with the primary structure.

For HSCT4.0, two mass cases are considered during multidisciplinary analysis for each geometric configuration of the aircraft: cruise weight and gross takeoff weight (GTOW). Typically, the aircraft center of gravity is farther aft during supersonic cruise than during takeoff, to allow the aircraft to be trimmed at a small angle of attack and small tail deflection angle during supersonic cruise. This change in the aircraft mass distribution during flight needs to be considered when designing the airplane, since the resulting stresses and buckling loads change as well. The current configuration as-built weight can be determined by a correlation of information from three sources: 1) the as-built structural, nonstructural, systems, payload, and fuel nodal weights for the baseline geometric airplane and mass distribution cases, 2) the theoretical FEM section properties and computed nodal weights for the current geometric configuration, and 3) empirical as-built structural, non-structural, and systems weights for various geometric configurations. Currently, a simplifying assumption has been introduced to eliminate the dependence on the empirical weights (the least reliable of the three sources); that is, the as-built nodal weight increments, due to nonstructural and systems weights, are assumed to be fixed, although these increments are allowed to move with geometric changes in the configuration. The takeoff gross weight is used as the objective function for the optimization.

Weight constraints are enforced to ensure that the operational empty weight is greater than zero, that the structural weight of each FEM component mesh is greater than zero, and that the fuel weight in each fuel tank is nonnegative. If the nonstructural and systems weights were allowed to change, additional constraints would be needed.

Nonlinear Correction Process

The *Nonlinear Correction* process is the first stage in what is called a variable-fidelity aerodynamic analysis approach. For efficiency during a design cycle, this approach uses only one computationally intensive, nonlinear CFD calculation per load condition. A nonlinear correction is then calculated relative to an appropriate linear aerodynamics calculation. In the second stage of the approach, this correction is applied many times during the *Loads Convergence* process. The nonlinear aerodynamic code used in HSCT4.0, the CFL3D code, has been widely used for aerodynamic analysis on a variety of configurations. Although the CFL3D code is capable of solving either the Euler or Navier-Stokes equations, for the HSCT4.0 application the code is used to solve the Euler equations to limit computational time in the HSCT4.0 application. Skin friction drag is accounted for in the *Polar* process.

The initial design cycle uses the baseline CFD surface grid, but subsequent design cycles use a surface grid that has been updated both for the changes in the design variables and also for the changes to the calculated displacements for each load condition in the *Loads Convergence* process. Once a small number of design cycles has been completed, it is expected that the changes to the calculated displacements between subsequent design cycles will be small, resulting in a consistent outer mold line shape for both the nonlinear and the linear aerodynamics calculations. The nonlinear aerodynamic surface modification from the MASSOUD code is input to the grid deformation code (CSCMDO)[33] to update the volume grid used in the CFD analysis. After the CFL3D code calculation has been made for each load condition, the pressure distribution is transferred to the panels of the linear aerodynamics grid by using a process that maintains the same total normal force and pitching moment. The linear aerodynamics code USSAERO is then run at an angle of attack that results in the same total normal force. The nonlinear correction is computed as the panelwise difference between the nonlinear pressure distribution and the linear pressure distribution.

Because the nonlinear correction is computed for a matching normal force, there is no net normal force contributed when the correction is later applied, but there will be a net pitching moment for the configuration; this moment is accounted for in the *Rigid Trim* process.

***Rigid Trim* Process**

The *Rigid Trim* process (see Fig. 7) represents the second stage in the variable-fidelity aerodynamic analysis approach. The purpose of the *Rigid Trim* process is to determine the configuration angle of attack and the tail deflection angle that combine to yield a lift that is equal to the weight, with no net pitching moment. A series of linear aerodynamic calculations are performed at combinations of angle of attack and tail deflection angle that bracket the expected range of conditions. The resulting surface pressures are then augmented by the nonlinear corrections before calculating total force and moment. The configuration angle of attack and the tail deflection angle are determined by linearly interpolating the USSAERO code calculations for the target lift coefficient and zero pitching moment. Lastly, the surface pressures are determined from the augmented surface pressures with the same linear combination of conditions.

***Polars* Process**

For the *Polars* process, the 1g cruise shape is used for all the aerodynamic calculations. The cruise result from the *Rigid Trim* process is augmented by calculating a set of induced drag coefficients for a range of Mach numbers and lift coefficient values to provide input to the calculations in the *Performance* process. At each Mach number for which the USSAERO code calculations are made, a range of angles of attack and two tail deflection angles are used. The resulting induced drag is interpolated at the lift coefficients appropriate for the FLOPS code input. The drag polars are obtained by combining these lift-induced drag contributions with the lift-independent drag contributions resulting from the skin friction, wave drag, and other miscellaneous drag increments calculated by other special-purpose codes. Nonlinear corrections are not used in the *Polars* process.

***Performance* Process**

The *Performance* process uses the FLOPS code to calculate the range and several other performance constraints needed for the optimization. The range is constrained to be greater than or equal to 5000 nautical miles. The balanced takeoff field length over a 35-foot high obstacle, including one engine out and aborted takeoff analyses, is constrained to be less than or equal to 10,000 feet. Similarly, the landing field length over a 50-foot high obstacle is also constrained to be less than or equal to 10,000 feet. The approach speed is constrained to be less than or equal to 155 knots. The time to climb to cruise is constrained to be less than or equal to 1 hour. Takeoff noise for flyover, sideline, and a combined metric are constrained to be less than or equal to that of the baseline configuration.

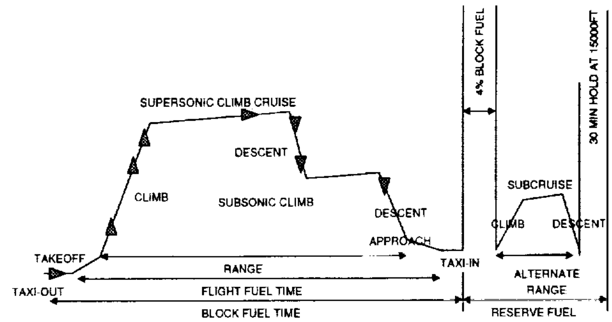


Fig. 11 Typical mission profile.

Figure 11 shows a typical mission profile for performance. The current geometric configuration, gross takeoff weight, wing fuel weights, fuselage fuel weights, aerodynamic data from the *Polars* process, and propulsion data for a reference aircraft are input to the FLOPS code. The code then solves the equations of motion for the input aircraft until a mission analysis consistent with the input geometry, weights, aerodynamics, and propulsion tables is obtained. The *Performance* process considers the takeoff, landing, climb, cruise, descent, and reserve portions of a specified mission profile, while requiring that the various Federal Aviation Regulations (FAR) and Federal Aviation Administration (FAA) flight regulations required for certification are satisfied. These regulations are summarized in Refs. 34–36.

***Ground Scrape* Process**

The *Ground Scrape* process provides constraints such that the aircraft tail will not scrape the ground on takeoff or landing. The ground scrape constraints are formulated as limits on the maximum values of the takeoff and landing gross weights; higher weights would require higher angles of attack, resulting in the aircraft tail scraping the ground.

Specifically, the *Ground Scrape* process computes maximum aircraft pitch angle to avoid tail strike (with a specified minimum ground clearance) for the landing gear just touching the ground at zero roll angle. The difference between the takeoff and landing conditions is that the landing gear is assumed to be at the static length for takeoff and at the fully stroked length for landing. The process also computes ground clearances for selected additional airframe and engine points at the same pitch angles and a given roll angle; these

clearances can be used to provide ground scrape warnings. After the *Ground Scrape* process calculates the maximum pitch angle, it executes the USSAERO code for the takeoff and landing conditions (assuming the angle of attack equals the ground scrape pitch angle) and extracts the lift coefficients. The process uses the takeoff and landing speeds to calculate the corresponding lift forces available based on the air density (from the standard atmosphere for a 95 ^{0}F day at 5000 feet above sea level) and the wing reference area.

The *Ground Scrape* process implemented here is a simple model of more realistic ground scrape processes that may be used by industry.

***Displacements* Process**

The *Displacements* process is used to generate structural deformations due to applied aerodynamic and weight loads. The first step in the *Displacements* process is the transformation of aerodynamic pressures on aerodynamic computational panels to aerodynamic forces at finite-element node locations in the z-direction by using the A2S code (see appendix). In the next step, the aerodynamic forces are augmented by the addition of the inertial loads (nodal weight vector times g-force) appropriate for that load condition. The GENESIS® code is then used to compute the structural deformations.

When the *Displacements* process is executed for the cruise load condition, a set of cruise displacements is generated. These displacements are saved as a reference set for use in the *Loads Convergence* process. The following assumption was applied to simplify the *Loads Convergence* process. Differences in the stiffness matrices of the cruise shape FEM and the unloaded shape FEM are assumed to be negligible. According to this assumption, the displacements on the cruise shape FEM will be identical to the displacements on the unloaded shape FEM when the same load is applied to each model. This "linear assumption for aeroelasticity" permits the use of the lofted cruise shape as the reference shape for both the aerodynamic and structural models. All finite-element analyses executed in the CJOpt system use the cruise shape FEM.

***Loads Convergence* Process**

The trimmed aerodynamic loads for each of the six noncruise load conditions are determined from an iterative aeroelastic analysis in the *Loads Convergence* process (Fig. 12). In the first step, the *Apply Delta Displacements* process uses a vector of "delta displacements" to perturb the shape of the derived linear aerodynamic grids generated by the *Geometry* process. Delta displacements are discussed below. For the first pass through the *Loads Convergence* loop, a vector of zero delta displacements is used, and the initial aerodynamic grids represent the cruise shape of the aircraft.

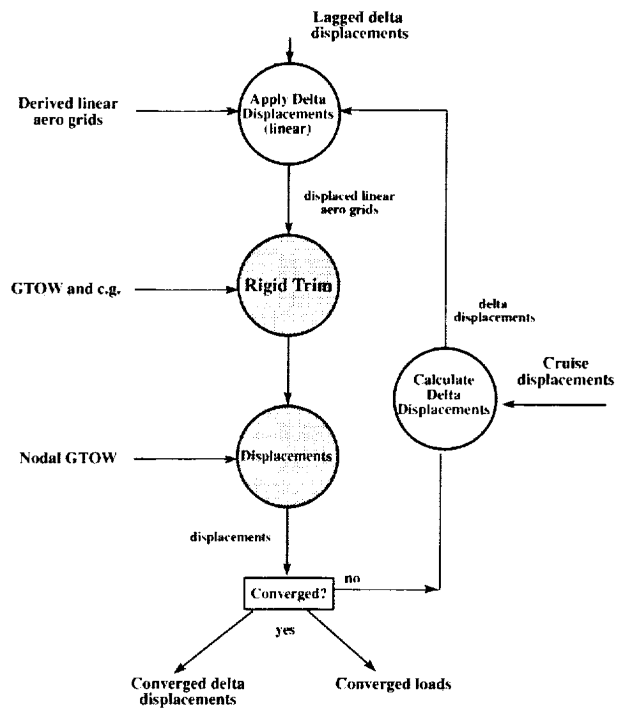


Fig. 12 *Loads Convergence* Process.

In the next step, the *Rigid Trim* process produces aerodynamic pressures, augmented by nonlinear corrections, which are transferred from the aerodynamic grid to the FEM grid for the current load condition. The weight vector is added to the aerodynamic load vector to produce a structural load vector. Then, the GENESIS® code uses this structural load vector to calculate displacements for the noncruise load conditions, as described in the *Displacements* process.

The *Loads Convergence* process continues until convergence. Convergence is achieved when the net vehicle shape being used for the aerodynamic calculations is consistent with the structural displacements caused by the aerodynamic loads. Typically, convergence is achieved in ten iterations.

The *Calculate Delta Displacement* process is only invoked if the convergence criterion is not met. For each load condition, this process computes the delta displacements as the displacements for that load condition minus the cruise displacements, as shown in Fig. 13. This process and the *Apply Delta*

Displacement processes are performed by the S2W code (see appendix).

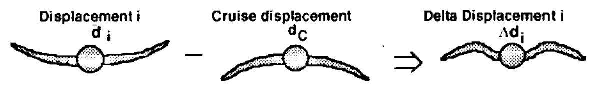


Fig. 13 Calculation of delta displacement for load condition "i"

Stress & Buckling **Process**

Stress analysis must be performed on the zero-stress (unloaded) shape of the FEM. However, all of the model grids generated by the *Geometry* process represent the aircraft configuration at the cruise load condition. According to the "linear assumption for aeroelasticity" (described in the *Displacements* process), a load vector applied to the cruise shape FEM will produce the same results (stress and displacement) as the same load vector applied to the unloaded shape FEM. Because of the "linear assumption for aeroelasticity," it is possible to use the cruise shape FEM for the stress analysis.

In the *Stress & Buckling* process, the stress and buckling constraints are obtained in the following manner. The six load conditions produced by the *Loads Convergence* process are added to a fuselage cabin pressure and the total is multiplied by a 1.5 factor of safety. The GENESIS® code uses these six augmented loads and a taxi load condition to compute stress failure indices and stress resultants.

For the stress constraints, only the maximum layerwise Hoffman[22] stress failure index (SFI) in each element is used. The Hoffman SFI is computed from the following equation:

$$\mathrm{SFI} = \sigma_x\left(\frac{1}{X_T}+\frac{1}{X_C}\right)+\sigma_y\left(\frac{1}{Y_T}+\frac{1}{Y_C}\right)-\frac{\sigma_x^2}{X_T X_C} - \frac{\sigma_y^2}{Y_T Y_C}-\frac{\tau_{xy}^2}{S^2}+\frac{\sigma_x\sigma_y}{X_C X_T}$$

For the buckling constraints, the inplane stress resultants computed in the GENESIS® code are used to calculate a buckling load factor (BLF) for each of the sized elements:

$$\mathrm{BLF} = \frac{N_1}{N_{mn}}+\left(\frac{N_{xy}}{N_{\mathrm{shear}}}\right)^2$$

The N_{mn} term in the above equation is obtained from the nontrivial solutions for stability of a simply-supported square plate under uniform biaxial compression[37] (the N_{mn} used is the smallest of the 25 combinations of m,n = 1 to 5 representing 25 buckling modes). The N_{shear} term in the BLF equation is obtained from the shear buckling interaction equation.[38]

A stress constraint and a buckling constraint are computed for each element and each load case in the 61 design zones on the fuselage and wing, shown in Fig. 3. This computation process yields an extremely large number of constraints. For example, in design zone 49, there are 28 elements. In this design zone, for the seven load conditions, there would be 196 stress constraints and 196 buckling constraints. The total number of structural constraints is 31,640. For optimization purposes, this large number of constraints could be reduced considerably by using a Kreisselmeier-Steinhauser[39] (KS) function to lump all the individual stress constraints into 1 KS stress constraint per zone and 1 buckling constraint per zone. This would result in 61 stress constraints and 61 buckling constraints. Alternatively, the KS function could be used to lump the individual stress and buckling constraints by load condition. This method would result in 427 stress constraints and 427 buckling constraints (one per load condition per zone). One of the goals of the HSCT4.0 project is to investigate how to handle the large number of stress and buckling constraints in the optimization.

Software Configuration Management

Because of its complexity, it was evident that the HSCT4.0 application development and associated CJOpt framework development required the use of formal procedures for software configuration management (SCM). This is the first purely research project at LaRC to use formal SCM methods. This section briefly discusses the motivation, implementation, and experience with SCM in the combined HSCT4.0-CJOpt project. Reference 40 describes in more detail the approach taken and experiences gained. It is hoped that the HSCT4.0-CJOpt project experience with SCM will be useful for other complex software research projects.

Motivation

Software configuration management (SCM) defines a set of methods and tools for identifying and controlling software during its development and use. Typical SCM activities include baseline establishment, change control and tracking, and reviews of the

evolving software. The application of SCM increases the reliability and quality of software. SCM is typically applied to the development of production software applications, such as business, control systems, and engineering, with clear requirements and a well-defined life cycle—it has rarely been used in software research.

The simple, manual SCM methods that had been used for configuration management during the development of the previous HSCT applications were inadequate. Some of the difficulties encountered were losing track of changes to the codes, poor handling of changes required by operating system updates, keeping insufficient records of the reasons for changes, and inconsistently applying version identifiers to software files. Consequently, additional work was required to reconstruct lost (or misplaced) versions when they were needed for testing new frameworks or communication systems.

The expected benefits of an SCM process for HSCT4.0 include consistent version control of each software item (code, test data, test procedure, or document), minimized risk of losing valuable information, clearly established roles and responsibilities, and assured ability to retrieve correct previous versions of software. Version control is particularly important because it helps to ensure all developers use consistent software versions.

Approach

An SCM Plan was developed for the HSCT4.0 project. The SCM Plan defines the methods and tools used for identifying and controlling the HSCT4.0 project software throughout its development and use. Specifically, it defines the SCM activities, how and when they are to be performed, who is responsible for each activity, and what resources are required. The Plan states that all HSCT4.0-CJOpt software products are to be placed under SCM; in addition to code, these products include makefiles, documentation, test case scripts, and test input and output. The Plan serves as a reference document for the project's SCM procedures. The Plan also includes sections on the schedule for implementing SCM, on the purpose and timing of functional and physical configuration audits, and on Plan maintenance.

The nature of the research environment in which the software is being developed was seriously considered while developing the SCM Plan. In the research environment, requirements necessarily evolve as the research progresses. Therefore, the Plan for HSCT4.0 and CJOpt was made to be more flexible than typical plans, and it is anticipated that the Plan will have to be adjusted to accommodate research necessities as experience is gained with SCM.

A combination of software tools is used to support the HSCT4.0 SCM activities. In the early stages of the Plan development, the TRUEchange™ ‡ product[41] of TRUE Software, Inc., was selected as the software tool for version control. An advantage of SCM software tools such as the TRUEchange product is the ability to maintain multiple operational versions of configuration items. Later, a set of Web-based electronic forms, including formal trouble reports, change requests, and promotion notifications, was selected to manage changes to the software. These electronic forms and the associated change-control metrics database had been developed earlier at LaRC and were adapted to meet the needs of the HSCT4.0-CJOpt project.

Experience

Even though the tools and processes have been only partially demonstrated, the HSCT4.0-CJOpt project team is finding that utilization of SCM is crucial for keeping track of the various modifications to codes. However, already there have been some problems in the project's SCM implementation. Some of the lessons learned so far from the experience in applying SCM to the HSCT4.0-CJOpt multidisciplinary optimization project are given below.

Early in the project, team members tended to bypass SCM procedures in an effort to save time. Because of these bad habits, the team experienced the inability to regenerate research results consistently and the unintentional use of multiple, inconsistent versions of a code. The problems experienced with software development when team members bypassed the SCM system have resulted in a greater acceptance of the need for SCM by the project team. Lesson learned: SCM must be consistently applied in order to reap the benefits.

To promote consistent application, the Plan and its implementers need to be specific in defining the procedures and responsibilities; templates and examples of what is expected have been helpful. Also, the subcontractor tasks must explicitly address the use of SCM. The HSCT4.0 MDO application involved a prolonged requirements analysis effort; SCM was introduced before design was complete. Lessons learned: the software design must progress to the point that software configuration items can be clearly

‡ The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

identified for version control before the configuration items can be defined successfully; the project schedule must allow adequate time to introduce SCM and to perform it.

Status

The *Analysis* process shown in Fig. 7 has been incorporated into CJOpt, a CORBA-compliant Java code software integration environment. Initially, the *Analysis* process required about 8 hours of wall clock time to execute sequentially on a heterogeneous mixture of high performance workstations, excluding the nonlinear aerodynamic code calculations. The nonlinear aerodynamic code would require 3 additional hours in parallel mode on an eight-processor workstation; all other codes run on single-processor machines. The *Analysis* process has been parallelized and now requires about 4 hours wall clock time on a heterogeneous mixture of high performance workstations with five iterations used in the *Loads Convergence* process—excluding the nonlinear aerodynamic code calculations. The sensitivity analysis and optimization phases are currently under development. A stand-alone nonlinear aerodynamic optimization, which uses the *Geometry* process, the nonlinear aerodynamics code (CFL3D), and the SLP optimizer, has been developed. The detailed results for the *Analysis* process and the stand-alone aerodynamic optimization process are discussed in a companion paper.[15]

Two sets of initial design variables are used to validate that the multidisciplinary analysis is integrated correctly. The term "integrated correctly" means that the values of the design variables and all quantities derived from the design variable values are passed from one process to another process correctly. The first set of design variable values known as the Baseline is based on the set of design variable values that correspond to the baseline FEM. When this set of design variable values is used the baseline results are reproduced. The second set of design variable values known as Higher Aspect Ratio (HAR) is based on a planform shape with a higher aspect ratio than the baseline and structural design variable values that are increased based on the following schema. If a design variable representing a 0° ply or a 45° ply is within 10 percent of its lower bound value, the value was increased by approximately 23 percent. If a design variable representing a 90° ply is within 10 percent of its lower bound value, the value is increased by approximately 145 percent. If a design variable representing the core is within 10 percent of its lower bound value, the value is increased by approximately 355 percent. The HAR design is expected to have an increased weight and changes in stress responses and buckling responses.

Acknowledgements

First and foremost, we thank our co-authors in fact (if not in name): R. P. Weston, B. H. Mason, L. L. Green, and R. T. Biedron. The authors also acknowledge the contributions of Computer Sciences Corporation to CJOpt integration; team members are R. Sistla, A. R. Dovi, P. Su, R. Shanmugasundaram, and R. Krishnan. Additionally, the authors acknowledge the helpful suggestions of former National Research Council fellow A. Giunta, who is now at Sandia National Laboratories, G. A. Wrenn of BA/NYMA, Inc, and the contributions of N. Alexandrov, H. J. Dunn, and T. A. Zang of the Multidisciplinary Optimization Branch at the NASA LaRC. The project was supported by the High Performance Computing and Communications Program, Computational AeroSciences office at NASA LaRC, lead by J. Sobieszczanski-Sobieski, D. H. Rudy, and J. J. Rehder. The configuration management effort was initially supported by the NASA Headquarters Software Process Improvement Activity and the LaRC Information Systems and Services Division, with guidance from M. P. Schuler and administrative support from Computer Sciences Corporation employees C. A. Malloy and K. L. Marlowe. The HSCT4.0 project is built on the previous HSCT projects, originally designed by T. M. Eidson of High Technology Corp. and implemented through the contributions of M. S. Adams, R. L. Barger, G. L. Giles, R. E. Gillian, D. P. Randall, and W. E. Zorumski of NASA LaRC, and of LaRC contractor employees M. A. Baddourrah, K. W. Edwards, R. L. Gates, D. P. Hammond, N. R. Hathaway, B. B. James, W. J. Lamarsh, T. Sreekantamurthy, and B. W. Wedan.

References

[1]Weston, R. P., Townsend, J. C., Eidson, T. M., and Gates, R. L., "A Distributed Computing Environment for Multidisciplinary Design," *Proceedings of the 5th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Part 2, Panama City, FL, 1994, pp. 1091–1097.

[2]Krishnan, R., Sistla, R., and Dovi, A. R., "High-Speed Civil Transport Design Using FIDO," NASA CR-1999-209693, Oct. 1999.

[3]Balabanov, V. O., Kaufman, M., Giunta, A. A., Haftka, R. T., Grossman, B., Mason, W. H., and Watson, L. T., "Developing Customized Wing Weight Function by Structural Optimization on Parallel Computers," *Proceedings of the 37th*

AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Part 1, Salt Lake City, UT, 1996, pp. 113–125.

[4]Giunta, A. A., Balabanov, V. O., Haim, D., Grossman, B., Mason, W. H., Watson, L. T., and Haftka, R. T., "Wing Design for a High-Speed Civil Transport Using a Design of Experiments Methodology," *Proceedings of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Part 1, Bellevue, WA, 1996, pp. 168–183.

[5]Giunta, A. A., and Sobieszczanski-Sobieski, J., "Progress Toward Using Sensitivity Derivatives in a High-Fidelity Aeroelastic Analysis of a Supersonic Transport," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Part 1, St. Louis, MO, 1998, pp. 441–453.

[6]Barthelemy, J.-F. M., Wrenn, G. A., Dovi, A. R., Coen, P. G., and Hall, L. E., "Supersonic Transport Wing Minimum Weight Design Integrating Aerodynamics and Structures," *Journal of Aircraft*, Vol. 31, No. 2, 1994, pp. 330–338.

[7]Hale, M., Craig, J. I., Mistree, F. and Schrage, D., "DREAMS and IMAGE: A Model and Computer Implementation for Concurrent, Life-Cycle Design of Complex Systems," *Concurrent Engineering: Research and Applications*, Vol. 4, No. 2, 1996, pp. 171–186.

[8]Guinta, A. A., "Sensitivity Analysis for Coupled Aero-Structural Systems," NASA/TM1999-209367, Aug. 1999.

[9]Sobieski, I. P., Manning, V. M., and Kroo, I. M., "Response Surface Estimation and Refinement in Collaborative Optimization," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Part 1, St. Louis, MO, 1998, pp. 359–370.

[10]Carlson, H. W., and Walkley, K. B., "Numerical Methods and a Computer Program for Subsonic and Supersonic Aerodynamic Design and Analysis of Wings With Attainable Thrust Considerations," NASA CR-3808, Aug. 1984.

[11]Giles, G. L., "Equivalent Plate Analysis of Aircraft Wing Box Structures with General Planform Geometry," *Journal of Aircraft*, Vol. 23, No. 11, 1986, pp. 859–864.

[12]Krishnan, R., and Eidson, T., "An Efficient Parallel Space-Marching Euler Solver for HSCT Research," *12th AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 1995, pp. 1189–1199.

[13]Stanley, G., Hurlbut, B., Levit, I., Loden, W., Stehlin, B., and Swenson, L., "Computational Mechanics Testbed With Adaptive Refinement User's Manual," Lockheed Palo Alto Research Laboratory, Palo Alto, CA, Feb. 1995.

[14]Weston, R. P., Green, L. L., Salas, A. O., Samareh, J. A., Townsend, J. C., Walsh, J. L., "Engineering Overview of a Multidisciplinary HSCT Design Framework Using Medium-Fidelity Analysis Codes," *1998 Computer Aerosciences Workshop*, NASA CP-20857, Jan. 1999, pp. 133–134.

[15] Walsh, J. L., Weston, R. P., Samareh, J. A., Mason, B. H., Green, L. L., and Biedron, R. T., "Multidisciplinary High-fidelity Analysis and Optimization of Aerospace Vehicles, Part 2: Preliminary Results," AIAA Paper 2000-0419, Jan. 2000.

[16]Sistla, R., Dovi, A. R., Su, P., and Shanmugasundaram, R., "Aircraft Design Problem Implementation Under the Common Object Request Broker Architecture," *40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, St. Louis, MO, 1999, pp. 1296–1305B.

[17]Sistla, R., Dovi, A. R., and Su, P., "A Distributed, Heterogeneous Computing Environment for Multidisciplinary Design & Analysis of Aerospace Vehicles," *5th National Symposium on LARGE-Scale Analysis, Design and Intelligent Synthesis Environments*, Oct 12–15, 1999, Williamsburg, VA.

[18]Vinoski, S., "CORBA: Integrating Diverse Applications Within Distributed Hetergeneous Environments," *IEEE Communications Magazine*, Vol. 14, No. 2, 1997, pp.46–55.

[19]Scotti, S. J., "Structural Design Using Equilibrium Programming Formulations," NASA TM-110175, June 1995.

[20]Woodward, F. A., "USSAERO Computer Program Development, Versions B and C," NASA CR-3228, Apr. 1980.

[21]Krist, S. L., Biedron, R. T., and Rumsey, C. L., "CFL3D User's Manual (Version 5.0)," NASA TM-208444, June 1998.

[22]Vanderplaats, G. N., *GENESIS User's Manual, Version 4.0*, Colorado Springs, CO, 1997.

[23]McCullers, A., "Aircraft Configuration Optimization Including Optimized Flight Profiles," *Proceedings of the Symposium on Recent Experiences in Multidisciplinary Analysis and Optimization*, edited by J. Sobieski, NASA CP-2327, Hampton, VA, April 1984, pp. 396–412.

[24]Vanderplaats, G. N., "CONMIN—A Fortran Program for Constrained Function Minimization, User's Manual," NASA TM-X-62282, Aug. 1973.

[25]Bischof, C., Carle, A., Corliss, G., Griewank, A., and Hovland, P., "ADIFOR—Generating Derivative Codes from Fortran Programs, " *Scientific Programming*, No. 1, 1992, pp. 1–29.

[26]Bischof, C., Carle, A., Khademi, P., and Mauer, A., "The ADIFOR 2.0 System for the Automatic Differentiation of Fortran 77 Programs," 1994. CRPC-TR94491.

[27] Bischof, C., Carle, A., Hovland, P., Khademi, P., and Mauer, A., "ADIFOR 2.0 User's Guide (Revision D)," CRPC-TR95516-S, March 1995, Rev June 1998.

[28]Bischof, C., Roh, L., and Mauer, A., "ADIC—An Extensible Automatic Differentiation Tool for ANSI-C, " Technical Report ANL/MCS-P626-1196, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Ave., Argonne, IL 60439, 1996.

[29]Samareh, J. A., "A Survey of Shape Parameterization Techniques," *AIAA/CEAS/ICASE/NASA-LaRC International Forum on Aeroelasticity and Structural Dynamics Conference*, edited by Woodrow Whitlow and Emily N. Todd, NASA / CP-1999-209136, Hampton, VA, 1999, pp. 333–343.

[30]Samareh, J. A., "A Novel Shape Parameterization Approach," NASA / TM-1999-209116, May 1999.

[31]Walatka, P. P., Buning, P. G., Pierce, L., and Elson, P. A., "PLOT3D User's Manual," NASA TM-1990-101067, Mar. 1990.

[32]Mitchell, P. M., "Advanced Finite Element Weight Estimation Process on the High Speed Civil Transport," SAWE Paper No. 2169, Presented at the 52nd Annual Conference of the Society of Allied Weight Engineers, Inc. May 24–25, 1993, Biloxi, MS.

[33]Jones, W. T., and Samareh-Abolhassani, J., "A Grid-Generation System for Multidisciplinary Design Optimization," *Proceedings of the 12th AIAA Computational Fluid Dynamics Conference*, San Diego, CA, 1995, pp. 657–669.

[34]Roskam, J., and Lan, C.-T. E., *Airplane Aerodynamics and and Performance*, Design, Analysis and Research Corporation (DARcorporation), Lawrence, KS, 1997.

[35] Roskam, J., *Airplane Design, Part I: Preliminary Sizing of Airplanes*, Design, Analysis and Research Corporation (DARcorporation), Lawrence, KS, 1997.

[36]Nicolai, L. M., *Fundamentals of Aircraft Design*, METS, Inc., Distributor, Xenia, OH, and E. P. Domincone Printing Services, Fairborn, OH, 1975.

[37]Whitney, James M., *Structural Analysis of Laminated Anisotropic Plates*, Technomic Publishing Company, Inc., Lancaster, PA, 1987, p. 104.

[38]Haftka, R. T., and Gürdal, Z., *Elements of Structural Optimization*, 3rd revised and expanded ed., Kluwer Academic Publishers, Boston, 1992, pp. 416–420.

[39]Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems*, Zurich, Switzerland, 1979, pp. 113–117.

[40]Townsend, J. C., Salas, A. O., and Schuler, M. P., "Configuration Management of an Optimization Application in a Research Environment," NASA / TM-1999-209335, June 1999.

[41]*TRUEchange Command Line User Interface, Version 2.1*, TRUE Software, Inc., Waltham, MA, 1997.

[42]Harder, R. L., and Desmarais, R. N., "Interpolation Using Surface Splines," *Journal of Aircraft*, Vol. 9, No. 2, 1972, pp. 189–191.

Appendix

This appendix provides a brief overview of the primary tools used in this research: the A2S code for load transfer, the ADIC code for differentiation, the ADIFOR code for differentiation, the CFL3D code for nonlinear aerodynamics, the CSCMDO code for grid deformation, the GENESIS® code for structural analysis, the MASSOUD code for shape parameterization, the FLOPS code for performance, the S2W code for deflection transfer, and the USSAERO code for linear aerodynamics. The GENESIS®, CFL3D, MASSOUD, FLOPS, and CSCMDO codes are capable of providing sensitivity derivatives. The GENESIS® code is a commercial

product and all other codes were developed by or for NASA.

A2S

The Aerodynamics-to-Structures, A2S, code transfers the aerodynamic loads to the structural elements using a distribution process that preserves the total normal (z-component) force and moment of the aerodynamic forces. The surface pressure on each aerodynamic panel is first converted into a single force normal to the panel at its center. Only the configuration normal component of each panel force is then distributed among the nodes of the closest structural element. The distribution is done so that the total normal force and the total x-moment and y-moment at the structural nodes is the same as that of the aerodynamic panel force. This process is repeated for all the aerodynamic panels to transfer all the aerodynamic loads to the structure.

ADIC[28]

The ADIC code is a tool for the automatic differentiation of C programs, loosely based upon methods and technology developed for the ADIFOR code. Given a C source code and a user's specification of dependent and independent variables, the ADIC code will generate an augmented derivative code that computes the partial derivatives of all of the specified dependent variables with respect to all of the specified independent variables, in addition to the original result.

ADIFOR[25-27]

The ADIFOR code is a tool for the automatic differentiation of FORTRAN77 programs. Given a FORTRAN77 source code and a user's specification of dependent and independent variables, the ADIFOR code will generate an augmented derivative code that computes the partial derivatives of all of the specified dependent variables with respect to all of the specified independent variables, in addition to the original result.

CFL3D[21]

The CFL3D code solves the three-dimensional, time-dependent Euler and thin-layer Navier-Stokes equations with a finite-volume formulation on structured grids. The equations are advanced in time implicitly with the use of 3-factor approximate factorization. It can employ grid sequencing, multigrid, and local time-stepping to accelerate convergence to steady state. It can also utilize a wide variety of grid multiple block connection strategies—including point matched, patched, and overset grid connections—in order to handle complex geometric configurations. Second-order upwind-biased spatial differencing is used for the inviscid terms, and flux limiting is used to obtain smooth solutions in the vicinity of shock waves. Viscous terms, if present, are centrally differenced. Several turbulence models of varying complexity are available. The particular version of the code used here is known as CFL3dv4.1hp. This version has been ported to parallel computer architectures via the use of MPI protocols. Furthermore, the automatic differentiation tool ADIFOR[32] has been applied to this version of the CFL3D code. The resulting code is able to provide a numerical solution to the Euler (or Navier-Stokes) equations as well as consistent derivatives of the numerical solution with respect to shape design variables.

CSCMDO[33]

The Coordinate and Sensitivity Calculator for Multidisciplinary Design Optimization (CSCMDO) code is a general purpose multi-block three-dimensional volume grid generator which is suitable for Multidisciplinary Design Optimization. The code is timely, robust, highly automated, and written in ANSI "C" for platform independence. Algebraic techniques are used to generate and/or modify block face and volume grids to reflect geometric changes resulting from design optimization. Volume grids are generated/modified in a batch environment and controlled via an ASCII user input deck. This allows the code to be incorporated directly into the design loop. Volume grids have been successfully generated/modified for a wide variety of configurations.

FLOPS[23]

The Flight Optimization System (FLOPS) code is a multidisciplinary system of computer programs for conceptual and preliminary design and evaluation of advanced aircraft concepts. It consists of nine primary modules: Weights, Aerodynamics, Engine cycle analysis, Propulsion data scaling and interpolation, Mission performance, Takeoff and landing, Noise footprint, Cost analysis, and Program control.

The FLOPS code may be used to analyze a point design, parametrically vary certain design variables, or optimize a configuration with respect to these design variables (for minimum gross weight, minimum fuel burned, maximum range, minimum cost, or minimum NO_x emissions) using nonlinear programming techniques. The configuration design variables are wing area, wing sweep, wing aspect ratio, wing taper ratio, wing thickness-chord ratio, gross weight, and thrust (size of engine). The performance design variables are cruise Mach number and maximum cruise altitude. The engine cycle design variables are the design point turbine entry temperature, the maximum turbine entry temperature, the fan pressure ratio, the overall pressure ratio, and the bypass ratio for turbofan and turbine bypass engines. The aircraft

configuration, engine cycle and size, and the flight profile may be optimized simultaneously.

GENESIS®[22]

The GENESIS® code is a fully integrated finite-element analysis/design software package. Analyses are available for static, normal modes, direct and modal frequency analysis, and heat transfer. Shape, sizing and topology optimization are the design options available to the user.

MASSOUD[30]

The MASSOUD code is a parameterization tool for complex shapes suitable for a multidisciplinary design optimization application. The approach consists of three basic concepts: 1) parameterizing the shape perturbations rather than the geometry itself, 2) exploiting Soft Object Animation algorithms used in computer graphics, and 3) relating the deformation to aerodynamics shape design variables such as thickness, camber, twist, shear, and planform. The MASSOUD code formulation is independent of grid topology, and that makes it suitable for a variety of analysis codes such as CFD and CSM. The analytical sensitivity derivatives are available for use in a gradient-based optimization. This algorithm is suitable for low-fidelity (e.g., linear aerodynamics and equivalent laminated plate structures) and high-fidelity analysis tools (e.g., nonlinear CFD and detailed finite-element modeling).

S2W

The Structures-to-Wavedrag, S2W, code transfers the computed displacement from the structures grid to the linear aerodynamic grid. The transfer is accomplished by infinite-plate splines. This method is based on a superposition of the solutions for the partial differential equation of equilibrium for an infinite plate. The details of the method can be found in Ref. 42.

USSAERO[20]

The Unified Subsonic and Supersonic Aerodynamic analysis (USSAERO) code is a linear aerodynamic panel code that has incorporated a symmetrical singularity method to provide surface pressure distributions on a fuselage and wings in subsonic and supersonic flow. This method extends the range of application of the program to include the analysis of multiple engine nacelles or finned external stores. In addition, nonlinear compressibility effects in high subsonic and supersonic flows are approximated by using a correction based on the local Mach number at panel control points.